

UNIVERSIDADE DO MINHO
DEPARTAMENTO DE INFORMÁTICA

DESENVOLVIMENTO DE APLICAÇÕES WEB

JOSÉ CARLOS RAMALHO

Turnera
Rede Social de Alunos para Alunos

Filipe Monteiro (a80229) João Vilaça (a82339)
Leonardo Neri (a80056)

2 de Fevereiro de 2020

Conteúdo

1	Introdução	2
2	Planeamento e Análise do Projeto	2
2.1	Requisitos	2
2.2	Planeamento	2
2.3	Arquitetura	3
3	Modelo de Dados	4
4	API	6
4.1	Autenticação	6
4.2	Endpoints	6
5	Interface	8
6	Conclusão	9

1 Introdução

O desenvolvimento deste trabalho surge no âmbito da unidade curricular de Desenvolvimento de Aplicações WEB, pertencente ao primeiro semestre do 4º ano/1º ano do curso de MiEI. O objetivo deste trabalho prático é implementar uma aplicação que sirva um propósito semelhante ao de uma rede social de alunos para alunos onde seja possível partilhar materiais, discutir datas, combinar eventos, devendo por isso possuir uma interface gráfica pública que pode ser utilizado por qualquer utilizador autenticado. Além disso, foi também criada uma API de dados que permite a gestão dos dados inerentes à utilização e gestão da aplicação.

2 Planeamento e Análise do Projeto

2.1 Requisitos

Após a consulta e análise do enunciado, foram retirados e definidos os seguintes requisitos para o sistema.

Requisitos essenciais:

- Gestão de utilizadores
- Publicação de conteúdos

Requisitos básicos:

- Distinção público/privado
- Comentar Publicações
- Gestão de ficheiros
- Gestão de grupos

Requisitos extra:

- Pedidos de amizade
- Feed de um utilizador
- Notificações
- Pesquisa de entidades

2.2 Planeamento

O primeiro passo, depois de definidas as tecnologias a utilizar, para ajudar no desenvolvimento, será preparar ferramentas que facilitem o trabalho, nomeadamente usar Docker para a infraestrutura (bases de dados) e criar scripts de povoamento de dados.

Passada esta fase de preparação, a equipa deverá focar-se primeiramente em construir um API básica que suporte as funcionalidades essenciais do sistema, gestão de utilizadores e publicações.

Quando estes primeiros endpoints estiverem funcionais, deverá repartir-se a força de trabalho, e enquanto algumas pessoas ficam responsáveis por avançar com os restantes endpoints da API, outras deverão começar a desenvolver a interface.

2.3 Arquitetura

No fim, da análise de requisitos, é fundamental avaliar e escolher as tecnologias usadas nas várias componentes dos sistema.

Base de dados: Neo4j, devido ao poder e facilidade na gestão, criação e análise de bases de dados maioritariamente caracterizadas pelas fortes conexões entre entidades. (Relacionamentos entre utilizadores, utilizadores com publicações e grupos, entre grupos e publicações, etc...)

API: Node.js, permite um desenvolvimento rápido de aplicações, imensas bibliotecas disponíveis através do NPM e é executado no Google Chrome's V8 runtime que é rápido, eficiente e facilmente escalável.

Interface: React.js, é fácil de aprender, dá grande poder no controlo do fluxo dos dados através de 'Data bindings' e é altamente rápido e escalável. Existe ainda uma imensa comunidade de suporte visto ter sido desenvolvido e apoiado pelo Facebook.

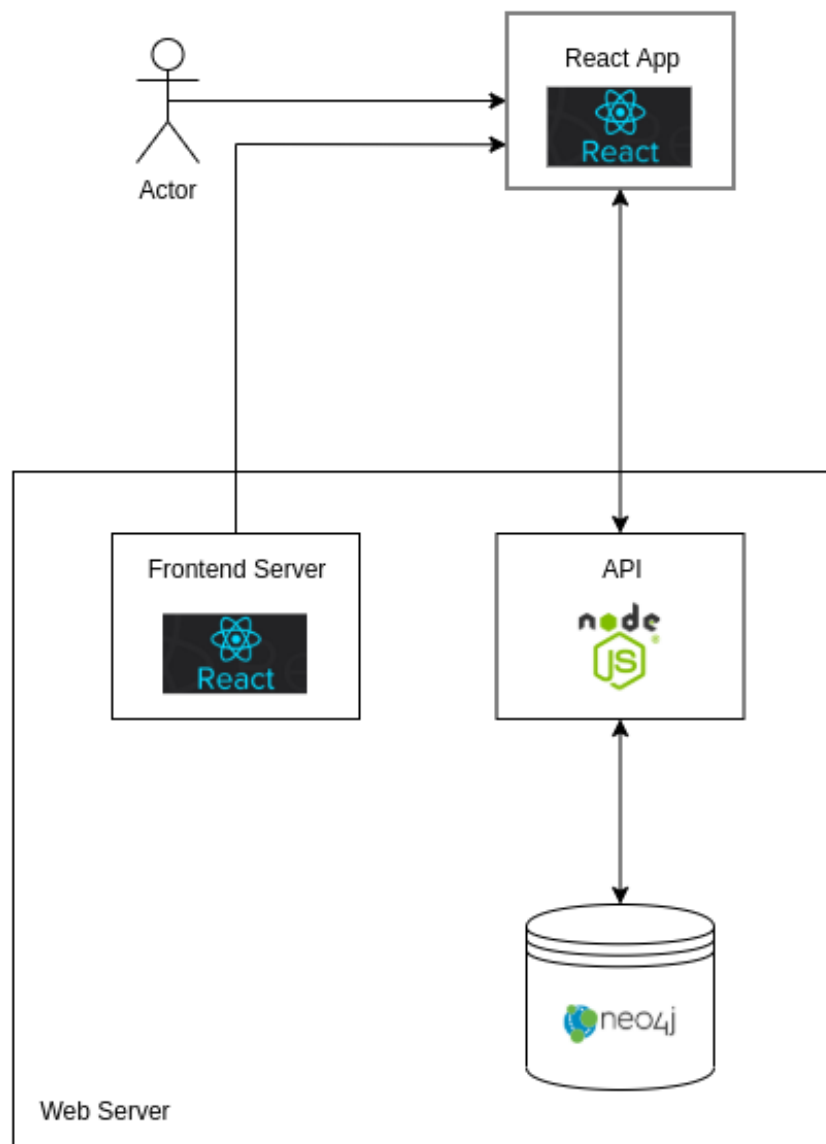


Figura 1: Arquitetura Geral

3 Modelo de Dados

O primeiro passo, antes de começar efetivamente o desenvolvimento, foi definir o seguinte modelo de dados capaz de suportar e gerir todo o processo de lógica de negócio que derivam dos requisitos do sistema.

User:

```
{
  "id": { type: String, unique: true },
  "email": { type: String, unique: true },
  "gender": { type: String, enum: ['Male', 'Female', 'Other'] },
  "name": { type: String },
  "home_town": { type: String },
  "lives_in": { type: String },
  "image": { type: String },
  "token": { type: String }
}
```

Friend:

```
{
  "user1": { type: String },
  "user2": { type: String },
  "since": { type: Date }
}
```

Friend Request:

```
{
  "user": { type: String },
  "invited": { type: String }
}
```

Groups:

```
{
  "name": { type: String, unique: true },
  "users": [String]
}
```

Posts:

```
{
  "id": { type: String, unique: true },
  "text": { type: String },
  "hashtags": [String],
  "owner": { type: String },
  "group": { type: String },
  "likes": [String],
  "images": [String]
}
```

Comments:

```
{  
  "post": { type: String },  
  "user": { type: String },  
  "text": { type: String }  
}
```

4 API

Nesta secção, iremos apresentar os 'endpoints' disponibilizados pela API que permitem satisfazer todos os requisitos do sistema.

4.1 Autenticação

A autenticação é feita recorrendo a dois passos distintos.

Existem dois endpoints abertos, '/api/signin' e '/api/signup', que se destinam ao início de sessão e criação de utilizadores. Estes quando bem sucedidos, email e palavra-passe corretos e utilizador criado respetivamente, é devolvido um token JWT, válido durante 2 horas, que deve ser usado para todos os pedidos efetuados aos restantes endpoints.

Neste segundo passo, em cada pedido efetuado à API é verificado que o token é válido, e este é ainda usado para identificar o utilizador, pelo que os resultados de cada pedido são específicos para as permissões desse mesmo utilizador.

4.2 Endpoints

```
/api/signin POST -- Início de sessão de um utilizador
/api/signup POST -- Registar um novo utilizador

/api/users -- Operações sobre utilizadores
  /:user_id
    GET -- Obter
    PUT -- Atualizar
    DELETE -- Apagar

/api/groups -- Operações sobre grupos
  /:group_id
    GET -- Obter
    PUT -- Atualizar
    DELETE -- Apagar
  / POST -- Criar

/api/users/:user_id/groups -- Operações sobre relações Utilizador-Grupo
  / GET -- Listagem dos grupos a que pertence
  /:group_id
    GET -- Obter
    POST -- Entrar em grupo
    DELETE -- Sair de grupo

/api/users/:user_id/posts -- Operações sobre relações Utilizador-Publicação
  /
    GET -- Listar
    POST -- Criar
  /:post_id
    GET -- Obter
    DELETE -- Apagar

/api/users/:user_id/friends -- Operações sobre amizades
  / GET -- Listar
  /:friend_id DELETE -- Remover

/api/users/:user_id/friend-requests -- Operações sobre pedidos de amizade
```

```

    / GET    -- Listar
    /:friend_id
        POST -- Enviar
        DELETE -- Cancelar/Recusar

/api/groups/:groups_id/posts -- Operações sobre relações Grupo-Publicação
    /
        GET -- Listar
        POST -- Criar
    /:post_id
        GET -- Obter
        DELETE -- Apagar

/api/posts/:post_id/like
    / POST -- Gostar de uma publicação

/api/posts/:post_id/comment
    / POST -- Criar comentário numa publicação

/api/search -- Operações de pesquisa
    /users POST -- sobre utilizadores
    /posts POST -- sobre publicações
    /groups POST -- sobre grupos

```


5 Interface

Login e Registo: primeira página apresentada, na qual o utilizador pode iniciar a sua sessão ou registar-se.

Pesquisa: o utilizador pode introduzir um termo na barra de pesquisa da navbar, e quando submete, um Modal aparece com os resultados da pesquisa.

Grupos: nesta página, o utilizador tem a oportunidade de ver os grupos a que pertence. Caso esteja a ver a página de um grupo ser-lhe-á apresentado um botão para entrar no grupo (caso não pertença) ou as publicações desse grupo.

Utilizadores: existem várias páginas e nuances para os utilizadores, caso o utilizador atual seja amigo de outro, poderá ver os seus posts, senão poderão ser apresentadas informações sobre possíveis pedidos de amizade existentes ou um botão para enviar pedido de amizade.

Perfil: no perfil, um utilizador vê os seus post e pode editar as suas informações

Publicação: na publicação, o utilizador pode consultar o texto e as hashtags dessa publicação, a informação do utilizador que fez a publicação e pode ainda ver o comentários da mesmo e até criar ele um novo comentário.

6 Conclusão

A realização deste projeto possibilitou o uso e aprendizagem das várias tecnologias leccionadas, utilizando conceitos e ferramentas desenvolvidas nas aulas e nos trabalhos para casa, o que permitiu desenvolver habilidades relacionadas com o planeamento, desenho e desenvolvimento de interfaces gráficas web e interfaces RESTful de dados.

Por fim, consideramos ter atingido a maioria dos requisitos propostos para este projeto, e adicionalmente foram desenvolvidas algumas funcionalidades extras que representam uma grande melhoria na usabilidade e utilidade do sistema. As limitações em termos de tempo foram a principal dificuldade, não tendo dado a oportunidade de melhorar alguns pormenores a nível gráfico que dariam uma ainda melhor experiência de utilização aos clientes.